Three-dimensional UAS Trajectory Optimization for Remote Sensing in an Irregular Terrain Environment

Youngjun Choi[1], Younghoon Choi[2], Simon Briceno[3], and Dimitri N. Mavris[4]

***Abstract*— This paper presents a novel algorithm for three-dimensional UAS trajectory optimization for a remote sensing mission in an irregular terrain environment. The algorithm consists of three steps: terrain modeling, the selection of scanning waypoints, and trajectory optimization. The terrain modeling process obtains a functional model using a Gaussian process from terrain information with a point cloud. The next step defines scanning waypoints based on the terrain model information, sensor specifications, and the required image resolution. For the selection of the waypoints, this paper introduces two different approaches depending on the direction of the viewing angle: a normal offset method and a vertical offset method. In the trajectory optimization, the proposed algorithm solves a distance-constraint vehicle routing problem to identify the optimum scanning route based on the waypoints and UAS constraints. Numerical simulations are conducted with two different UAS trajectory scanning methods in a realistic scenario, Point Loma in San Diego.**

I. INTRODUCTION

In the last decade, Unmanned Aerial Systems (UAS) have become more capable platforms because of more advanced technologies such as new battery technologies, light weight structures, upgraded sensor systems and novel autopliot algorithms. Hence, UAS has been applied to various areas (e.g., 3D mapping, disaster monitoring, precision agriculture, border patrol, search and rescue, and building inspection). In particular, 3D mapping is a core research field since many applications typically use high-resolution images or 3D mapping results.

For a 3D mapping mission, defining a flight coverage path is a challenge because of limited battery life that constrains flight endurance time. The typical endurance range of a Commercial Off-The Shelf (COTS) quadcopter is approximately between 10 and 30 minutes, and the endurance range of a COTS fixed wing aircraft is approximately between 30 minutes and 2 hours. In order to scan a large coverage area, the flight path must be efficiently designed to satisfy the endurance constraint of a given UAS platform.

Many trajectory generation algorithms have been introduced to identify an optimum trajectory. Notable trajectory optimization algorithms can be divided into five categories: stochastic approaches, road map methods, potential field methods, geometric methods, and optimization-based approaches [5]. In particular, grid-based trajectory generation is well-suited for a coverage trajectory optimization problem since the required an image resolution can determine the size of grid cells that define coverage waypoints. In aerial imaging missions, popular grid-based trajectory generation methods are as follows, classical exact cellular decomposition, wavefront-based algorithms, and vehicle routing problem-based optimization techniques.

The classical exact cellular decomposition generates a sweeping trajectory to cover an entire Area of Interest (AOI), which applies a zigzag route on discretized cells [8]. This sweeping method is computationally fast, but it has a limitation when an AOI is a non-convex shape. To solve this limitation, decomposition techniques that divide an AOI into multiple convex areas have been introduced. The representative decomposition methods are trapezoidal, the boustrophedon, and Morse-based cellular decompositions [1]. The trapezoidal decomposition creates multiple trapezoids or triangles using an extended vertical line at each vertex. However, the drawback of this method is that it generates many small sub-areas. Therefore, in order to specify the optimal routing sequence for visiting all the sub-areas, it requires an inefficient and complicated high-level optimization scheme. As a result, using this method may require an additional function that merges small areas to reduce the number of sub-areas. To mitigate this issue, the boustrophedon decomposition has been introduced, which decomposes a scanning area using critical vertexes, but this method also has a limitation when it has non-polygon restricted areas or obstacles inside of an AOI [6]. The Morse-based cellular decomposition efficiently solves the non-polygon restricted area issue through generating a more smooth scanning trajectory depending on the selection of a Morse function [1].

An alternative grid-based method is using a wavefront-based algorithm, a well-known coverage trajectory technique in the field of robotics [18][8]. This method applies a wave propagation algorithm that assigns numbers to each grid within an AOI based on initial/terminal positions and the information of restricted areas. Using the assigned numbers of each grid, pseudo-gradient information is employed to compute a complete coverage trajectory. The advantage of this method is that it is able to solve a non-convex AOI with a non-linear objective function. Hence, the wavefront-based algorithm has been applied to solve an optimum UAS scanning trajectory problem [3][12].

*This work was not supported by any organization

[1]Research Engineer, School of Aerospace Engineering, Georgia Institute of Technology, ychoi95@gatech.edu

[2]Graduate Researcher, School of Aerospace Engineering, Georgia Institute of Technology, younghoon.choi@gatech.edu

[3]Senior Research Engineer, School of Aerospace Engineering, Georgia Institute of Technology, briceno@gatech.edu

[4] S.P. Langley Distinguished Regents Professor, School of Aerospace Engineering, Georgia Institute of Technology, dimitri.mavris@aerospace.gatech.edu

Another grid-based trajectory optimization method is a vehicle routing-based approach proposed by George Dantzig et al., which solves an optimal route problem for vehicles from central depots to a set of customer locations [7]. The vehicle routing problem typically solves a cost function minimizing total traveling distance/time subject to one or multiple depots, a set of vehicles, the locations of customers, and customers' demands. The vehicle routing approach has a flexible structure that enables one to efficiently manage design variables such as the number of vehicles, fixed/free depots, and a set of vehicle constraints. For instance, this vehicle routing problem-based trajectory optimization scheme has been applied to address the UAS coverage problem [2].

Most of recent literature associated with a coverage path-planning algorithm is handling a 2D terrain problem that generally assumes a flat surface. In other words, path-planning algorithms generate a complete scanning trajectory on Above Ground Level (AGL) that does not actually account for the shape of the ground surface. In agriculture robot applications, Hameed et al. have stated that ignoring elevation changes is a too optimistic assumption because of the significant elevation impact [9]. It implies that the coverage trajectory of an aerial image also needs to consider characteristics of the terrain topology. In a 3D UAS scanning trajectory area, few articles have addressed this issue of the 3D coverage trajectory, mostly inspection missions such as building and structures [10] [4].

This work proposes a three-dimensional UAS trajectory optimization algorithm for a remote sensing mission to capture the actual terrain's topological characteristics, which allows a more realistic coverage trajectory. The proposed method incorporates a Gaussian Process (GP)-based terrain modeling method and a distance-constrained vehicle routing problem. The terrain modeling process creates a terrain model using a GP-based on the information of a Digital Elevation Model (DEM). Then, using the GP terrain model, the proposed method determines UAS waypoints. Next, the scanning trajectory optimization solves a distance-constrained vehicle routing problem for an optimal scanning trajectory that must visit all the waypoints.

In the remainder of this paper, we introduce the proposed methodology for a three-dimensional coverage trajectory optimization algorithm. To be more specific, Section II-A describes the terrain modeling method using a GP, Section II-B presents waypoints selection based on the terrain model, and Section II-C introduces the formulation of the distance-constrained vehicle routing problem. Section III demonstrates the implementation using a representative case. Section IV provides concluding remarks.

II. New Three Dimensional UAS Scanning Trajectory

A. Sparse Gaussian Process-based terrain modeling

Large-scale terrain modeling is a highly challenging problem because of unstructured terrain data, incomplete information, and inherent sensor uncertainty. A common terrain modeling approach is applying surface triangulation, but this approach is not suitable to handle the uncertainty of terrain data. To address these types of issues, many techniques and methods with averaging and interpolation techniques have been introduced. One example method is that Hameed et al. applies a bilinear interpolation approach that uses the distance-weighted average using proximity points to specify a new point [9]. Another popular terrain modeling approach is Gaussian Process-based terrain modeling. The representative example of a GP-based terrain model is suggested by Vasudevan et al., which employs a local approximation method using KD-Trees for a scalable terrain model [17].

In this work, we apply a Gaussian Process for a terrain model because a GP regression is a powerful tool to handle uncertainties. A GP as a non-parametric technique is a collection of random variables, which have a finite number of subsets with a Gaussian distribution [13]. The GP model can be represented by

$$f(x) \sim \mathbf{GP}(\mu(x), k(x, \hat{x})), \tag{1}$$

where $\mu(x)$ is the mean function, and $k(x, \hat{x})$ is the covariance function.

It is assumed that we have a set of training data, $\{(\mathbf{x}_k, y_k)|i = 1, ..., n\}$, $(\mathbf{x} \in \mathbb{R}^{n\times m}, y \in \mathbb{R}^{n\times 1})$. In the terrain modeling problem, $\mathbf{x}$ is a vector of points in the xy axis, and y is z-axis information. All training input is denoted as $\mathbf{x}$ and the corresponding output is denoted as y. The output y is the result of adding Gaussian noise to a true function $\tilde{f}(\mathbf{x})$, which is $y_k = \tilde{f}(\mathbf{x}_k) + \epsilon_k$. Note that the Gaussian noise is defined as $\left(\epsilon \sim N(0, \sigma_n^2)\right)$. The $\mathbf{GP}$ regression model computes the posterior distribution that is obtained by the joint probability model, which is $p(\mathbf{y}, \tilde{f}) = p(\mathbf{y}|\tilde{f})p(\tilde{f})$. We note that $p(\mathbf{y}|\tilde{f})$ is the likelihood and $p(\tilde{f})$ is the prior distribution. The equations of the $\mathbf{GP}$ regression model, which is the posterior distribution, can be written as

$$\mu(\mathbf{x}) = \mathbf{K}(\hat{\mathbf{x}}, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n \mathbf{I})^{-1}\mathbf{y} \tag{2}$$

$$Cov(\mathbf{x}, \hat{\mathbf{x}}) = \mathbf{K}(\hat{\mathbf{x}}, \hat{\mathbf{x}}) - \mathbf{K}(\hat{\mathbf{x}}, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n \mathbf{I})^{-1}\mathbf{K}(\hat{\mathbf{x}}, \hat{\mathbf{x}}), \tag{3}$$

where $\mathbf{K}(\cdot)$ is the $n \times n$ covariance matrix, also called the kernel matrix, $\mathbf{x}$ is the training input, $\hat{\mathbf{x}}$ is the test input, and $\mathbf{y}$ is the training output.

In this paper, the kernel matrix is assumed to be a Gaussian kernel, also named the square exponential covariance function:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_s^2 exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right) + \sigma_n^2, \tag{4}$$

where $\mathbf{W}$, σ_s, and σ_n are the hyper-parameters. The hyper-parameters can be obtained by maximizing the log marginal likelihood given by

$$\begin{aligned}\Theta^* &= \underset{\Theta}{\operatorname{argmin}} \ \ log\left(p\left(\mathbf{y}|\mathbf{x}, \Theta\right)\right) \\ &= \underset{\Theta}{\operatorname{argmin}} \ \left\{-\frac{1}{2}\mathbf{y}^T(\mathbf{K}_y)^{-1}\mathbf{y} - \frac{1}{2}log|\mathbf{K}_y| - \frac{n}{2}log(2\pi)\right\},\end{aligned} \tag{5}$$

where the kernel matrix $\mathbf{K}_y$ is $\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n \mathbf{I}$

The GP technique is a flexible class of non-parametric probabilistic models. However, because of a non-parametric feature, the GP technique incurs a computational issue for a large dataset problem since it requires the scale of a matrix inversion as $O(n^3)$ time, where n is the number of training data points [15]. To solve this computational limitation, many sparse techniques have been introduced, which reduces the computational cost to $O(n^2m)$ time, where m is the number of pseudo-input points or inducing variables. The representative sparse GP methods are the Sparse Pseudo-input GP method (SPGP) and Projected Process Approximation (PP) [14][15]. However, these two methods change the GP prior distribution because of the approximation of the true covariance, which does not yield a robust approximation of the exact GP result. To address this issue, Titsias suggested a variation learning method to select inducing variables and optimize hyperparameters [16] and demonstrated robust regression result of the variation learning-based sparse GP through a comparison assessment with SPGP and PP. Therefore, we opt for the variation learning-based sparse GP as a terrain modeling method. The posterior mean and variance of the sparse GP method are as follows

$$\tilde{\mu}(\mathbf{x}) = \mathbf{K}(\mathbf{x},\tilde{\mathbf{x}})\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})^{-1}\left(\sigma^{-2}\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\mathbf{W}\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\mathbf{y}\right) \tag{6}$$

$$\tilde{Cov}(\mathbf{x},\hat{\mathbf{x}}) = \mathbf{K}(\hat{\mathbf{x}},\hat{\mathbf{x}}) - \mathbf{K}(\mathbf{x},\tilde{\mathbf{x}})\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\mathbf{K}(\tilde{\mathbf{x}},\mathbf{x}) \tag{7}$$

$$+ \mathbf{K}(\mathbf{x},\tilde{\mathbf{x}})\chi\mathbf{K}(\tilde{\mathbf{x}},\mathbf{x}), \tag{8}$$

where $\chi = \mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})^{-1}\left(\sigma_n^{-2}\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\mathbf{W}\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\mathbf{y}\right)\mathbf{K}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})^{-1}$

To optimize the hyperparameters, which are the variational quantities $\tilde{\Theta}^* = [\tilde{\mathbf{x}},\tilde{\Theta}]$, the sparse GP solves the following log marginal likelihood function.

$$\tilde{\Theta}^* = \underset{\tilde{\Theta}}{\operatorname{argmin}} \left\{-\frac{1}{2}\mathbf{y}^T(\tilde{\mathbf{K}}_y)^{-1}\mathbf{y} - \frac{1}{2}log|\tilde{\mathbf{K}}_y| - \frac{n}{2}log(2\pi)\right\}$$

$$+ \frac{1}{2\sigma_n^2}tr(\tilde{Cov}), \tag{9}$$

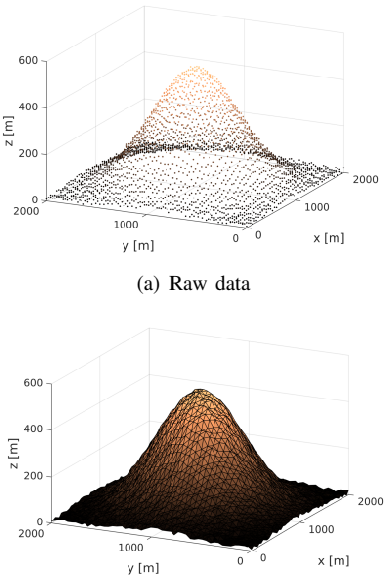
where $\tilde{\mathbf{K}}_y = \tilde{\mathbf{K}}(\mathbf{x},\tilde{\mathbf{x}})\tilde{\mathbf{K}}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})^{-1}\tilde{\mathbf{K}}(\tilde{\mathbf{x}},\mathbf{x}) + \sigma_n\mathbf{I}$, and $\tilde{Cov} = \tilde{\mathbf{K}}(\mathbf{x},\mathbf{x}) - \tilde{\mathbf{K}}(\mathbf{x},\tilde{\mathbf{x}})\tilde{\mathbf{K}}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})^{-1}\tilde{\mathbf{K}}(\tilde{\mathbf{x}},\mathbf{x})$. The variable $\tilde{x}$ is a subset of the training inputs. In the formulation, the last trace term is for a regularization to prevent overfitting. We note that the detailed derivation and explanation of the variation learning-based sparse GP is available in the literature [16].

To demonstrate the sparse GP-based terrain model, we generate Gaussian terrain data with a scaling factor as follows

$$T(\mathbf{x}) = \frac{K}{\sqrt{(2\pi)|\Sigma|}}exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

$$+ kN(0,\sigma^2), \tag{10}$$

where K is the scaling factor for the terrain model, k is the scaling factor for Gaussian noise, μ is the mean vector, and Σ is the covariance matrix. From the Gaussian terrain model, we collect raw terrain data as point cloud information, and obtain the approximate terrain model using the sparse-GP shown in Fig. 1.



(a) Raw data

(b) GP model

Fig. 1: GP model

B. Selection of scanning waypoints

The waypoints for a UAS scanning trajectory can be determined by the Field of View (FOV), the ground coverage of a single picture. The FOV, which is a function of image resolution and the specification of an optical sensor, defines the discrete grid space of an AOI that must be acquired from scanned images. The centers of these grid cells determine the scanning waypoints.

The size of an aerial image is the ratio of the focal length of a sensor (e.g., Charge-Coupled Device camera, or a Complementary Metal-Oxide Semiconductor camera) and an operational altitude, and the ratio of the pixel size of a sensor and the ground sampling distance:

$$S = \frac{f}{H} = \frac{C_x}{G_y} = \frac{C_y}{G_y}, \tag{11}$$

where f and H are the focal length and the operational altitude above ground level (AGL), and C_x, C_y indicate x and y pixel size of the sensor. G_x and G_y are x and y ground sampling distance, respectively. Note that H is also called offset distance in this paper. The Ground Sampling Distance (GSD), which corresponds the resolution of an image is

$$GSD = S \times P_{area}, \tag{12}$$

where P_{area} is the pixel area, which is $P_{area} = S_{area}/S_{resolution}$. In general, the stereo viewing and processing of an aerial image require one to consider endlap and sidelap for photogrammetric processing. The endlap as forward overlap is an overlapped area on two successive images along the flight direction. The sidelap as side overlap is an overlapped area of two aerial images perpendicular to the flight direction. Mathematically, the definition of both overlaps can be written by

$$\tilde{G}_x = (1-\gamma_x)G_x \tag{13}$$

$$\tilde{G}_y = (1-\gamma_y)G_y, \tag{14}$$

where γ_x and γ_y are the ratio of endlap $\tilde{G}_x$ and sidelap $\tilde{G}_y$. These overlap variables $\tilde{G}_x$ and $\tilde{G}_y$ can determine the size of grid cells, which defines the discretization of the AOI. Instead of using two overlap variables, we employ $min(\tilde{G}_x, \tilde{G}_y)$ to define the shape of a grid cell because the flight direction of a quadcopter to scan the AOI may not always be the same back-and-forth direction.

To determine 3D flight waypoints, offset distance and view direction also need to be considered. For the offset distance H, this is determined by the sensor and the image resolution GSD. For the view direction, this paper applies two different methods: vertical offset and normal offset.

The vertical offset method generates UAS scanning waypoints by a z-direction offset from the center of a surface patch. To be more specific, the viewing direction for the waypoints is the down direction in the local tangent plane, also called the North East Down (NED) coordinate system as the context of navigation. The waypoints $\mathbf{x}_w$ by the vertical offset method can be written by

$$\mathbf{x}_w = \mathbf{x}_c + [0 \ 0 \ H]^T, \tag{15}$$

where $\mathbf{x}_c$ is the center position of a grid cell.

The normal offset method generates UAS scanning waypoints by offset based on the surface gradient of a grid patch. To be more specific, this viewing direction is the normal vector of the center of the grid patch. The advantage of this normal viewing angle is that it is approximately perpendicular to the surface plane, which minimizes the distortion of an image. The normal vector can be computed by the partial derivative of a GP-based terrain model, $\mathbf{N} = [\frac{\partial \mu}{\partial \hat{\mathbf{x}}}, \frac{\partial \mu}{\partial \hat{\mathbf{y}}}, -1]$. The waypoints are the normal vector multiplied by the offset distance H, which is

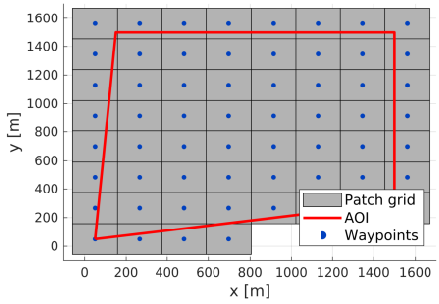
$$\mathbf{x}_w = \mathbf{x}_c + \mathbf{N} \ H \tag{16}$$

To demonstrate two different offset methods, we assume that the optical sensor is the Zenmuse X5S Olympus M.Zuiko 9-18mm. The specifications of the Zenmuse X5S are summarized in Table I. For the simplicity and better visual explanation of an example problem, we assume a coarse grid environment in which the required image resolution is 91.33 (cm), low resolution. Based on the sensor and GSD requirement, the operational altitude is 250 (m) and the size of the grid cells is 216 (m).

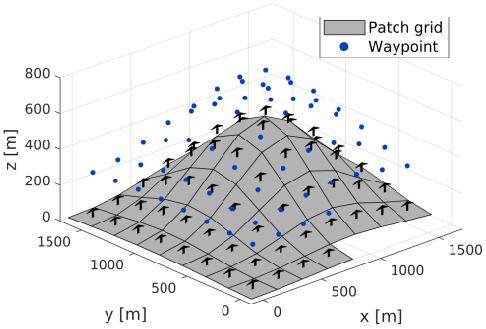
TABLE I: Specifications of Zenmuse X5S

Sensor width (mm)	17.3
Sensor height (mm)	13
Resolution (Mpix)	20.8
Lens Focal length (mm)	9

Based on the size of the grid cells, we discretize the AOI, the red polygon line in Fig. 2(a) and Fig. 3(a), and apply the two offset methods, as illustrated in Fig. 2 and Fig. 3. In the figures, arrows correspond to the offset direction, blue dots are waypoints, and the red line indicates the AOI. In the vertical offset method, as expected, the waypoints are located at the center of the grid patches. On the other hand, the waypoints resulting from the normal offset method are not located at the center of the grid patches because of the offset direction, which is the normal direction from the surface of each grid cell.



(a) Top view



(b) 3D view

Fig. 2: Waypoints based on the vertical offset method

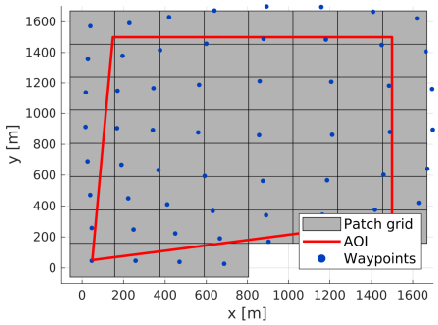
C. Distance constrained vehicle routing problem

Once the waypoints to scan the entire AOI are determined, the proposed framework solves a vehicle routing problem for an optimal scanning trajectory. In this vehicle routing problem, the framework used the distance constrained vehicle routing problem Kara suggested [11]. In the optimization formulation for the vehicle routing problem, we assume that the positions of initial/terminal depots are the same location and are known, vehicle speed is constant during a scanning mission, and the number of UAS platforms is one.

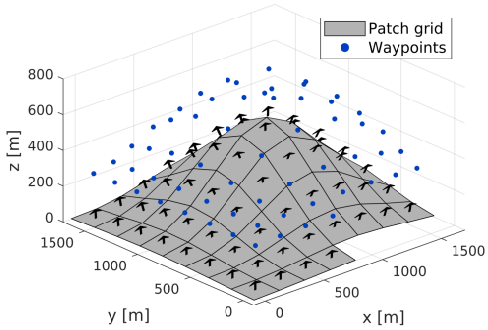
The general vehicle routing problem can be characterized by graph $G = (N, E)$ such that $N = D \cup C$, where N stands for nodes $(N = \{0, 1, ..., n\})$, D is a depot presented at node 0, and C is the set of n customers. The set E is the edge set, which is $E = \{(i, j) : i, j \in N, \ i \neq j\}$. We note that the edge means a route between two waypoints. The objective function of the vehicle routing problem is minimizing the travel distance

$$J = Min \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ij}, \tag{17}$$

where d_{ij} is a distance from the nodes i to j, and x_{ij} as a decision variable. If the edge (i, j) is selected, then $x_{ij} = 1$;



(a) Top view



(b) 3D view

Fig. 3: Waypoints based on the normal offset method

otherwise $x_{ij} = 0$, subject to

$$g_1 = \sum_{i=0}^{n} x_{0i} = 1 \tag{18}$$

$$g_2 = \sum_{i=0}^{n} x_{i0} = 1 \tag{19}$$

$$g_3 = \sum_{i=0}^{n} x_{ij} = 1 \quad (j = 1, 2, \cdots, n) \tag{20}$$

$$g_4 = \sum_{j=0}^{n} x_{ij} = 1 \quad (i = 1, 2, \cdots, n) \tag{21}$$

$$g_5 = \sum_{j=0, j\neq i}^{n} y_{ij} - \sum_{j=0, j\neq i}^{n} y_{ji} - \sum_{j=0} d_{ij} x_{ij} = 0, \quad (i = 1, 2, \cdots, n) \tag{22}$$

$$g_6 = y_{0i} = d_{0i} x_{0i}, \quad (i = 1, 2, \cdots, n) \tag{23}$$

$$g_7 = y_{ij} \leq (D - d_{j0}) x_{ij}, \quad j \neq 0, \quad (i, j) \in E \tag{24}$$

$$g_8 = y_{i0} \leq D x_{i0}, \quad (i = 1, 2, \cdots, n) \tag{25}$$

$$g_9 = y_{ij} \geq (d_{0i} + d_{ij}) x_{ij}, \quad i \neq 0, \quad (i, j) \in E \tag{26}$$

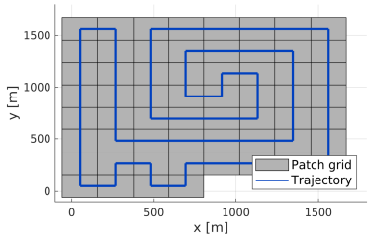
$$g_{10} = x_{ij} \in \{0, 1\}, \quad \forall (i, j) \tag{27}$$

Constraints g_1 and g_2 imply that a single vehicle departs from the depot (node 0), and arrives to the depot. Constraints g_3 and g_4 corresponds the degree constraints that each customer is visited once. Constraint g_5 is the subtour elimination constraint that prevents illegal subtours. Note that y_{ij} is the flow variable, the total distance from the depot to the node j when a vehicle travels from the nodes i and j. The constraints from g_6 to g_9 are the bounding constraints to guarantee $0 \leq y_{ij} \leq D$. In the constraint g_9, the variable x_{ij} is a binary variable.

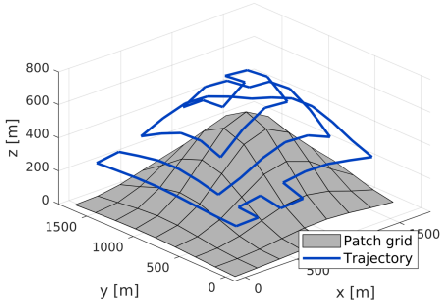
To test the formulated vehicle routing problem, we conducted an experiment with numerical simulations. In the experiment, we assumed that the UAS platform is the DJI Matrice 210. The cruise speed for the scanning mission is presumed to be $15(m/s)$ and, the maximum endurance for the scanning mission is assumed to be $20(min)$. We also assumed that the locations of the initial/final depots are at the center of a corner grid cell located at the bottom left side of Fig. 2(a). Based on the results of the two waypoints methods presented in Fig. 2 and Fig. 3, numerical simulation solves the trajectory optimization problem. Fig. 4 and Fig. 5 illustrate the results of the scanning trajectory optimization. Both methods cover 100 % of the waypoints. Table II summarizes the result of the scanning distance and time. The result illustrates that the scanning trajectory with the vertical offset method has less flight time than the scanning trajectory with the normal offset method. The proposed framework of the scanning trajectory optimization in an irregular terrain is summarized in Algorithms 1 and 2.

TABLE II: Result of an example study

	Algorithm 1	Algorithm 2
Total scanning distance (km)	13.29	14.77
Total scanning time (min)	14.77	15.66

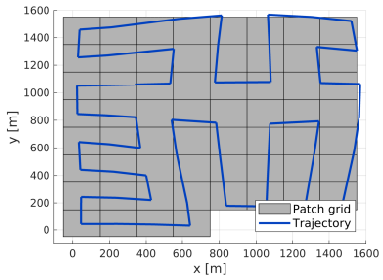


(a) Top view

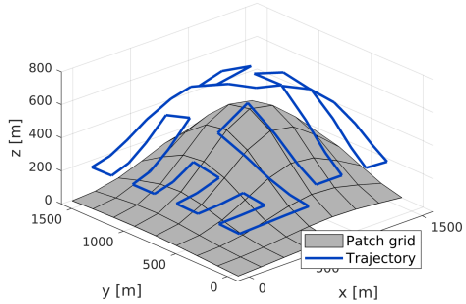


(b) Side view

Fig. 4: Results of distance constrained vehicle routing problem with the vertical offset method (Algorithm 1)



(a) Top view



(b) Side view

Fig. 5: Results of distance constrained vehicle routing problem with the normal offset method (Algorithm 2)

Algorithm 1 Optimal scanning trajectory with vertical offset approach

1: Sparse-GP based terrain modeling {(Section II-A)}
2: Selection of waypoints {(Section II-B, Eq. 15)}
3: Distance constrained-vehicle routing problem
4: {(Eq. 17 $\sim$ Eq. 27)}

III. NUMERICAL SIMULATION

In this section, the proposed UAS scanning trajectory algorithms are demonstrated in a representative terrain environment. To obtain a fully realistic terrain environment, we collected the 3D terrain information of a section of San Diego around Point Loma from a Digital Elevation Model (DEM) that consists of a point cloud, as shown in Figure 6. For the experiment, the optical camera sensor is assumed to be the Zenmuse X5S illustrated in Table I, and the UAS platform is assumed to be the DJI Matrice 210. We also assume that the requirement of the image resolution is assumed to be 0.04 (m) and both overlap rates are 0.4. The resulting operation altitude is approximately 120 (m) and the size of a grid cell is 104 (m).

Fig. 7 shows the results of the terrain scanning trajectory based on two proposed frameworks. The results clearly show that both scanning trajectories cover all grid cells, which must be passed. Table III summarizes the result of the scanning distance and time. The results show that both methods generate similar scanning distance/time.

Algorithm 2 Optimal scanning trajectory with normal offset approach

1: Sparse-GP based terrain modeling {(Section II-A)}
2: Selection of waypoints {(Section II-B, Eq. 16)}
3: Distance constrained-vehicle routing problem
4: {(Eq. 17 $\sim$ Eq. 27)}

TABLE III: Results of a numerical simulation in San Diego

	Algorithm 1	Algorithm 2
Total scanning distance (km)	10.49	10.40
Total scanning time (min)	11.66	11.55

IV. CONCLUSION

In this paper, we introduce a new framework for a three-dimensional coverage path-planning algorithm for a UAS terrain scanning problem. The framework consists of three steps: terrain modeling, waypoint selection, and vehicle routing problem-based trajectory optimization. In the terrain modeling, this paper applies the sparse-GP method using large terrain data from a digital elevation model. In the waypoint selection, the framework suggests two different offset methods depending on the UAS view direction, which are vertical offset and normal offset. In the trajectory optimization, the method employs the distance constrained vehicle routing problem. Numerical simulations with the proposed framework with the two different proposed offset methods are conducted and compared. The proposed method has a flexible structure, in which the terrain modeling can be accomplished with any of a myriad of terrain modeling methods, and the vehicle routing problem can be modified using a different cost function or constraints in the optimization. One potential extension of this framework is a multi-UAS coverage problem in an irregular terrain since in a large scanning area, multiple-UAS platforms may be required to scan the entire area.

REFERENCES

[1] Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002.

[2] Thiago Werlley Bandeira, Walton P. Coutinho, Alisson V. Brito, and Anand Subramanian. Analysis of path planning algorithms based on travelling salesman problem embedded in UAVs. In *Computing Systems Engineering (SBESC), 2015 Brazilian Symposium on*, pages 70–75. IEEE, 2015.

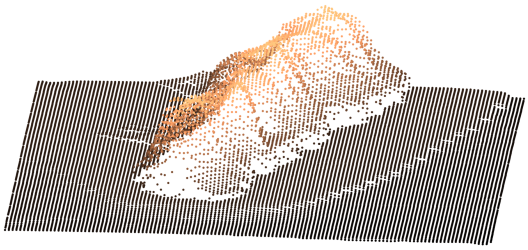
[3] Antonio Barrientos, Julian Colorado, Jaime del Cerro, Alexander Martinez, Claudio Rossi, David Sanz, and João Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5):667–689, 2011.

[4] Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6423–6430. IEEE, 2015.

[5] Youngjun Choi, Hernando Jimenez, and Dimitri N Mavris. Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories. *Robotics and Autonomous Systems*, 98:158–173, 2017.

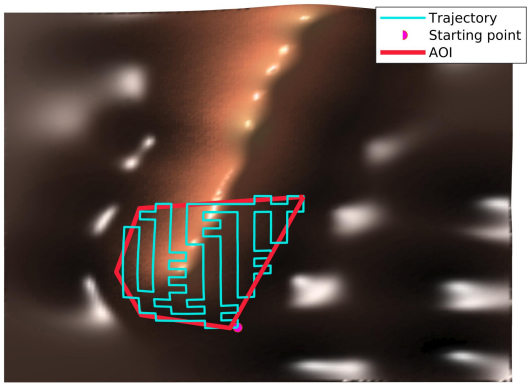
[6] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and service robotics*, pages 203–209. Springer, 1998.

(a) Point Loma in San Diego (Google Image)

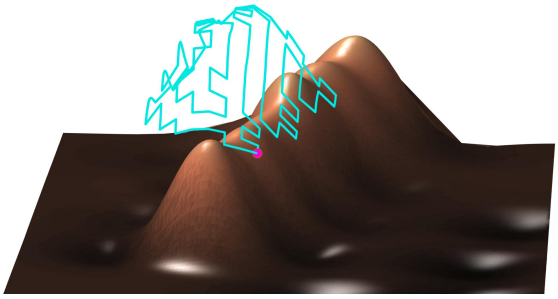


(b) Point cloud from Digital Elevation Model (DEM)

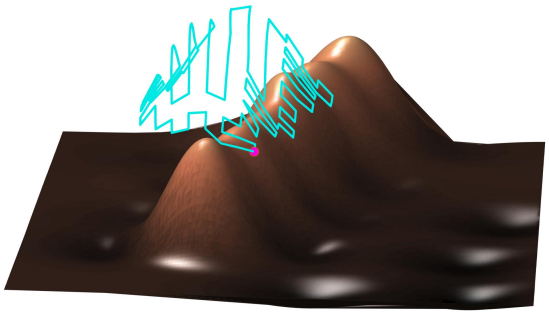
Fig. 6: Results of distance constrained vehicle routing problem



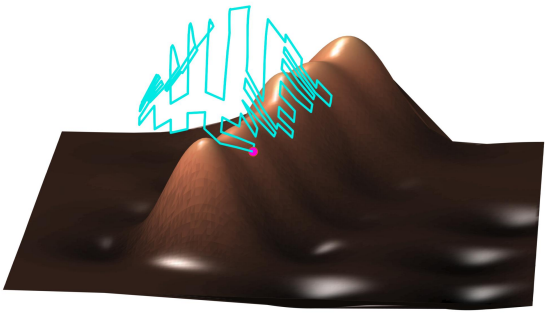
(a) Top view (Algorithm 1)



(b) 3D view (Algorithm 1)



(c) Top view (Algorithm 2)



(d) 3D view (Algorithm 2)

Fig. 7: Results of distance constrained vehicle routing problem

[7] George B. Dantzig and John H. Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[8] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[9] Ibrahim A. Hameed, Anders la Cour-Harbo, and Ottar L. Osen. Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. *Robotics and Autonomous Systems*, 76:36–45, 2016.

[10] Wei Jing, Joseph Polden, Wei Lin, and Kenji Shimada. Sampling-based view planning for 3D visual coverage task with unmanned aerial vehicle. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1808–1815. IEEE, 2016.

[11] Imdat Kara. Arc based integer programming formulations for the distance constrained vehicle routing problem. In *Logistics and Industrial Informatics (LINDI), 2011 3rd IEEE International Symposium on*, pages 33–38. IEEE, 2011.

[12] L.H. Nam, L. Huang, X.J. Li, and J.F. Xu. An approach for coverage path planning for UAVs. In *Advanced Motion Control (AMC), 2016 IEEE 14th International Workshop on*, pages 411–416. IEEE, 2016.

[13] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1.

[14] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318, 2003.

[15] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

[16] Michalis Titsias. Variational learning of inducing variables in sparse

Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[17] Shrihari Vasudevan, Fabio Ramos, Eric Nettleton, and Hugh Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840, 2009.

[18] Alexander Zelinsky, Ray A. Jarvis, J.C. Byrne, and Shinichi Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of international conference on advanced robotics*, volume 13, pages 533–538, 1993.